Laboratorio di

Sicurezza Informatica

Chiavi crittografiche

Marco Prandini

Dipartimento di Informatica – Scienza e Ingegneria

Ciclo di vita delle chiavi

- Generazione
  - Robustezza
- Memorizzazione
- Distribuzione

Generazione delle chiavi

- Per le chiavi simmetriche, i nonce, i vettori di inizializzazione, i padding, …
  - basta un buon generatore di numeri casuali
- Per le chiavi asimmetriche
  - servono numeri primi
  - si parte da numeri random
  - si applica un test di primalità

https://en.wikipedia.org/wiki/Primality_test

PRNG

- La casualità gioca un ruolo fondamentale per la generazione delle chiavi e la randomizzazione dei protocolli crittografici
- Che proprietà devono avere i numeri casuali?

Randomness

Unpredictability

Uniform distribution

→ The frequency of occurrence of ones and zeros should be approximately equal

Independence

→ No one subsequence in the sequence can be inferred from the others

TRUE random sequences

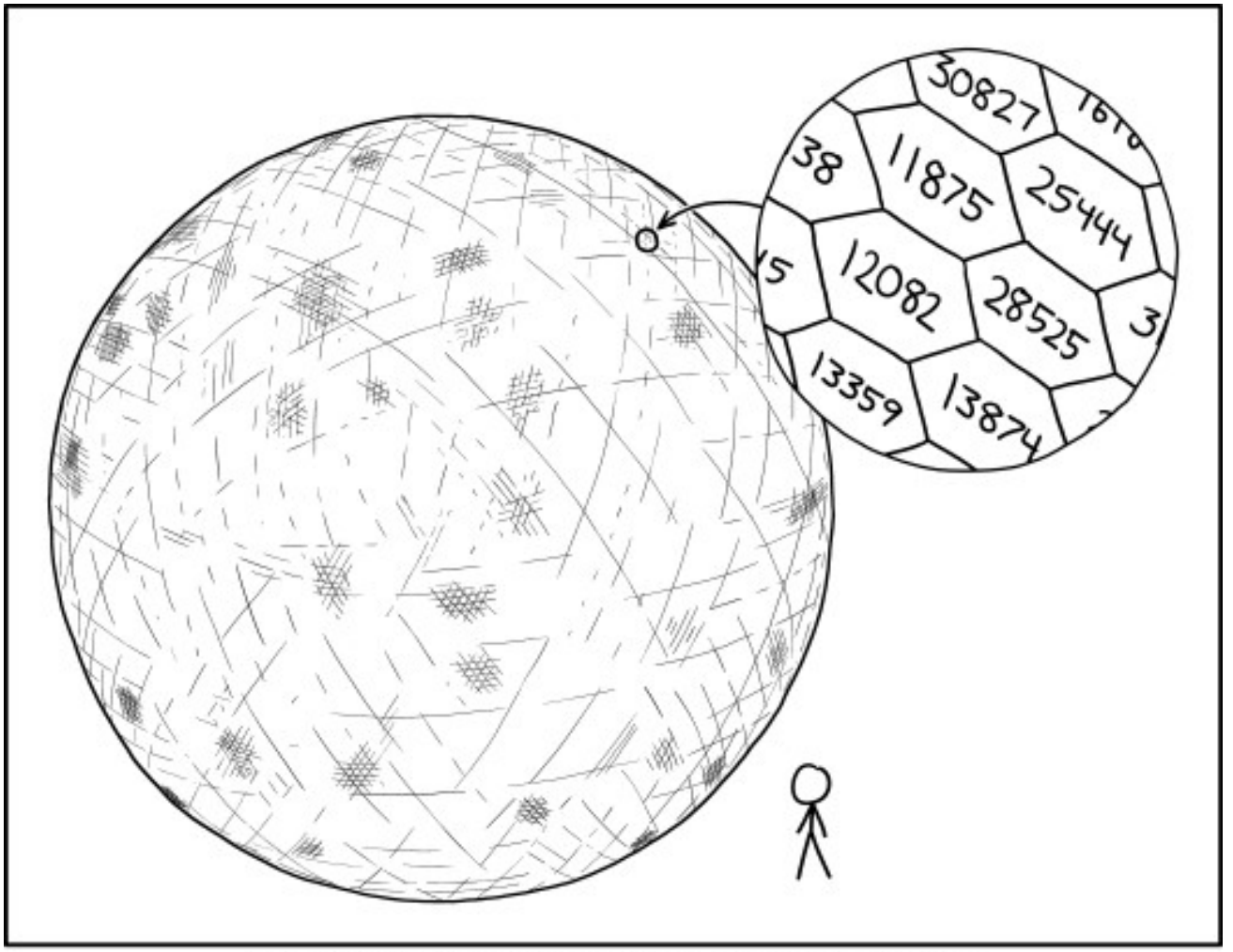
→ perfect independence guarantees unpredictability

HARD AND INEFFICIENT

PSEUDO random sequences

Algorithmic → must take extra care to make elements of the sequence hard to predict knowing earlier ones

True Random Number Generation



THE HARDEST PART OF SECURELY GENERATING RANDOM 16-BIT NUMBERS IS ROLLING THE D65536.

https://xkcd.com/2626/

True Random Number Generation

- Sorgenti **fisiche** di *entropia*
  - Elementi ad hoc, es. rumore termico, processi dinamici caotici
    https://blog.cloudflare.com/randomness-101-lavarand-in-production/
  - Eventi “imprevedibili” nel calcolatore, es. intervalli di arrivo degli interrupt dai dispositivi
- Elaborazione
  - Conversione A/D
  - Condizionamento (rimozione bias)

Pseudo Random Number Generation

- Algoritmo → Determinismo!
- Se il risultato supera i test statistici, accettabile come PRNG
  - Ma sempre attenzione all'imprevedibilità
- Tipicamente input = *seme* (*seed*) prodotto da TRNG
  - Noto il seme → nota la sequenza generata!
  - Se algoritmo robusto, una sequenza di valori intermedi non permette di risalire al seme o di ipotizzare il valori futuri

Test di casualità

- Standard NIST SP 800-22
  - 15 test

Frequency test

- The most basic test and must be included in any test suite
- Purpose is to determine whether the number of ones and zeros in a sequence is approximately the same as would be expected for a truly random sequence

Runs test

- Focus of this test is the total number of runs in the sequence, where a run is an uninterrupted sequence of identical bits bounded before and after with a bit of the opposite value
- Purpose is to determine whether the number of runs of ones and zeros of various lengths is as expected for a random sequence

Three tests

Maurer's universal statistical test

- Focus is the number of bits between matching patterns
- Purpose is to detect whether or not the sequence can be significantly compressed without loss of information. A significantly compressible sequence is considered to be non-random

Resistenza alla forza bruta

- Tempo di test dello spazio delle chiavi DES/AES con tecnologie recenti:

	Lunghezza della chiave in bit		
Budget	56	128	256
1 K€ (individuo)	16 anni	10^{22} anni	10^{61} anni
1 M€ (impresa)	6 giorni	10^{19} anni	10^{58} anni
1 G€ (NSA)	8 minuti	10^{16} anni	10^{55} anni

- Attenzione alle ricerche con tempo di calcolo gratis (lotteria cinese, virus) e alla sfortuna!
- Anche se la legge di Moore proseguisse, c'è un limite invalicabile: la termodinamica
  Limite di Landauer: per cambiare 1 bit almeno $k \times T \times \ln(2)$ (3×10^{-23} J a 3°K)
  Tutta l'energia emessa dal Sole in un anno = 1.2×10^{34} J
  → 4×10^{56} bit flip, come contare da 0 a 2^{188}
  Energia emessa dall'esplosione di una supernova = 2×10^{44} J
  → 7×10^{66} bit flip, come contare da 0 a 2^{222}

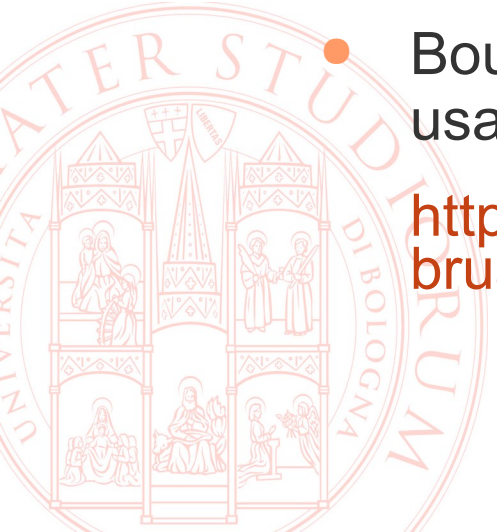
Resistenza alla fattorizzazione

- Il miglior attacco a RSA non è la forza bruta ma la ricerca dei fattori del modulo
- Stime non facilissime

  https://www.keylength.com/en/8/

  - 3000 bit robusti fino al 2026
- Stato attuale
  - Boudot et alii (2020) hanno fattorizzato RSA-250 (829 bit) usando 2700-anni-core

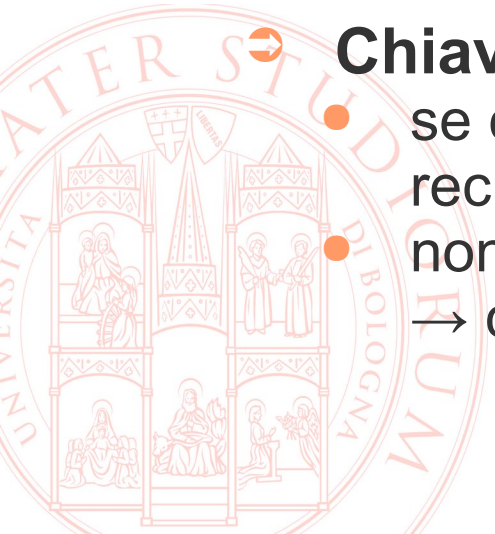
    https://lists.gforge.inria.fr/pipermail/cado-nfs-discuss/2020-February/001166.html



Memorizzazione

- **Chiave di decifrazione**: requisiti contrastanti
  - perdita devastante → backup
  - segretezza fondamentale → non diffusione
- Accorgimenti di memorizzazione
  - Cifratura con passphrase
  - Hardware Security Module
  - Key escrow
  - Secret sharing

- **Chiave di firma**
  - se compromessa si sostituisce, nessuno ha bisogno di recuperarla in assenza del titolare → nessun backup!
  - non deve essere usata contro la volontà del titolare → cifratura con passphrase, HSM



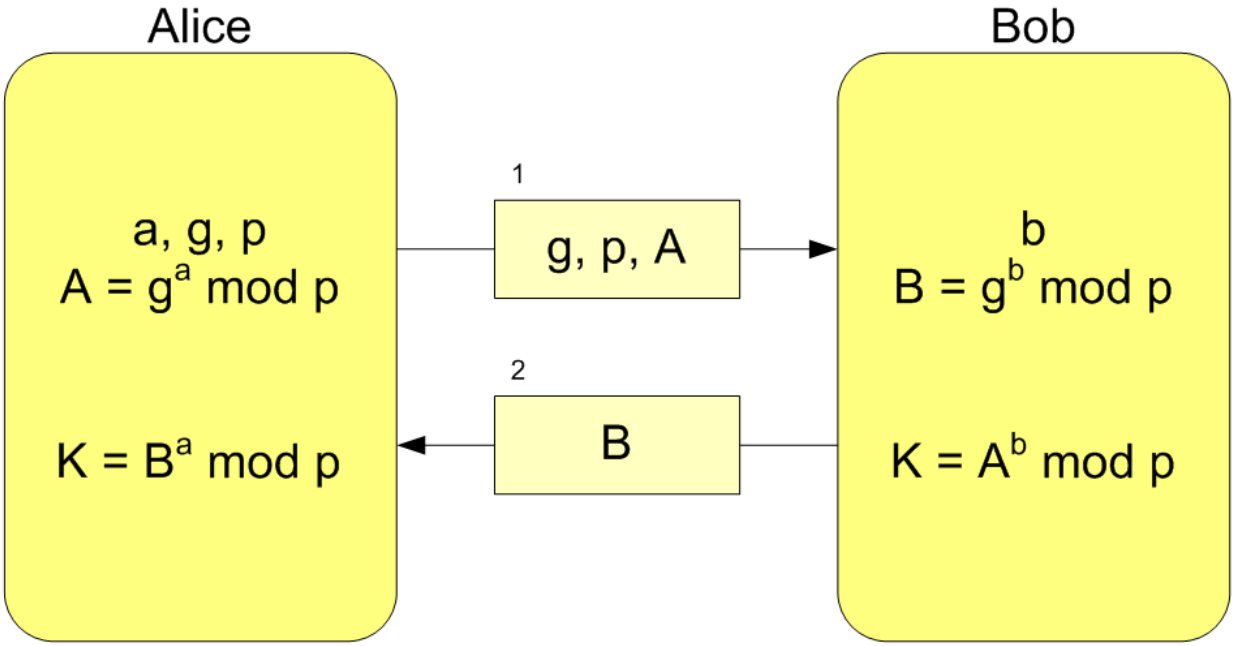
Gestione

- Altri parametri che complicano la gestione:
- Numero delle chiavi in gioco
  - Per i sistemi asimmetrici: una chiave pubblica per ogni soggetto
  - Per i sistemi simmetrici: una chiave segreta per ogni coppia di soggetti = N(N-1)/2 chiavi
- Aderenza a standard e policy
  - frequenza di sostituzione
  - compatibilità di formato
- OASIS KMIP

https://wiki.oasis-open.org/kmip/KnownKMIPImplementations

Distribuzione

- Chiavi simmetriche: non devono mai essere esposte in chiaro
  - scambio manuale
  - KDC
    - Ogni utente condivide una chiave con un centro fidato (Key Distribution Center)
    - Per stabilire una connessione, due utenti negoziano una chiave attraverso connessioni cifrate col KDC
  - Scambio di Diffie-Hellman →



$K = A^b \bmod p = (g^a \bmod p)^b \bmod p = g^{ab} \bmod p = (g^b \bmod p)^a \bmod p = B^a \bmod p$

Distribuzione

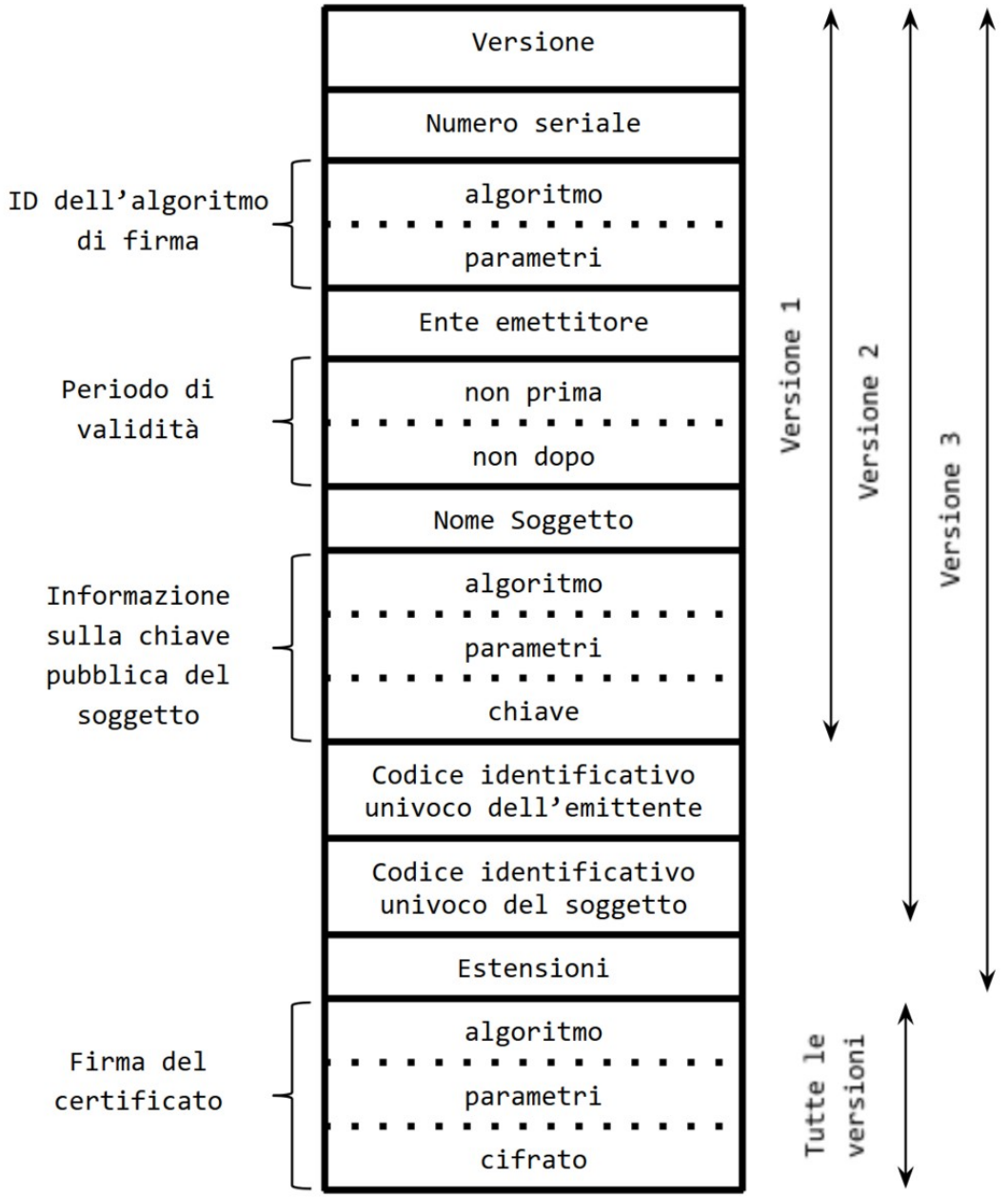
- Un attaccante passivo non apprende nulla dalla visione di una chiave pubblica o dall'intercettazione dei parametri di DH, **un attaccante attivo può invece sostituire i valori inviati da una parte all'altra coi propri**
- RSA: l'attaccante ha una propria coppia di chiavi $PRIV_I$ e PUB_i, e quando due utenti cercano uno la chiave pubblica dell'altro, ricevono invece PUB_I
  - quando il mittente cifra i messaggi, l'attaccante li può decifrare, e con la chiave pubblica del destinatario legittimo, per ri-cifrarli per non insospettirlo (magari alterati)
  - l'attaccante può firmare messaggi con $PRIV_I$ e il destinatario, verificandoli correttamente con $PRIV_I$ si convincerà che siano del mittente legittimo
- DH: l'attaccante stabilisce due chiavi separate con A e B e continua a fare da "passacarte" senza insospettirli
- **Il problema quindi per i sistemi asimmetrici non è la riservatezza, ma l'autenticità dei dati pubblici ricevuti**

Certificazione delle chiavi pubbliche

- Serve un modo per associare con certezza una chiave pubblica al suo legittimo titolare (nonché unico possessore della corrispondente chiave privata)
- Modello ***web of trust:***
  - L'autenticità di una chiave pubblica è testimoniata da altri utenti
  - L'utente che riceve una chiave da uno sconosciuto può decidere di accettarla per autentica se è firmata da qualcuno fidato
  - Vantaggio: nessuna entità "super partes" di cui doversi fidare
  - Svantaggio: pessima scalabilità
- Modello **infrastrutturale**:
  - esiste una terza parte fidata che documenta l'associazione

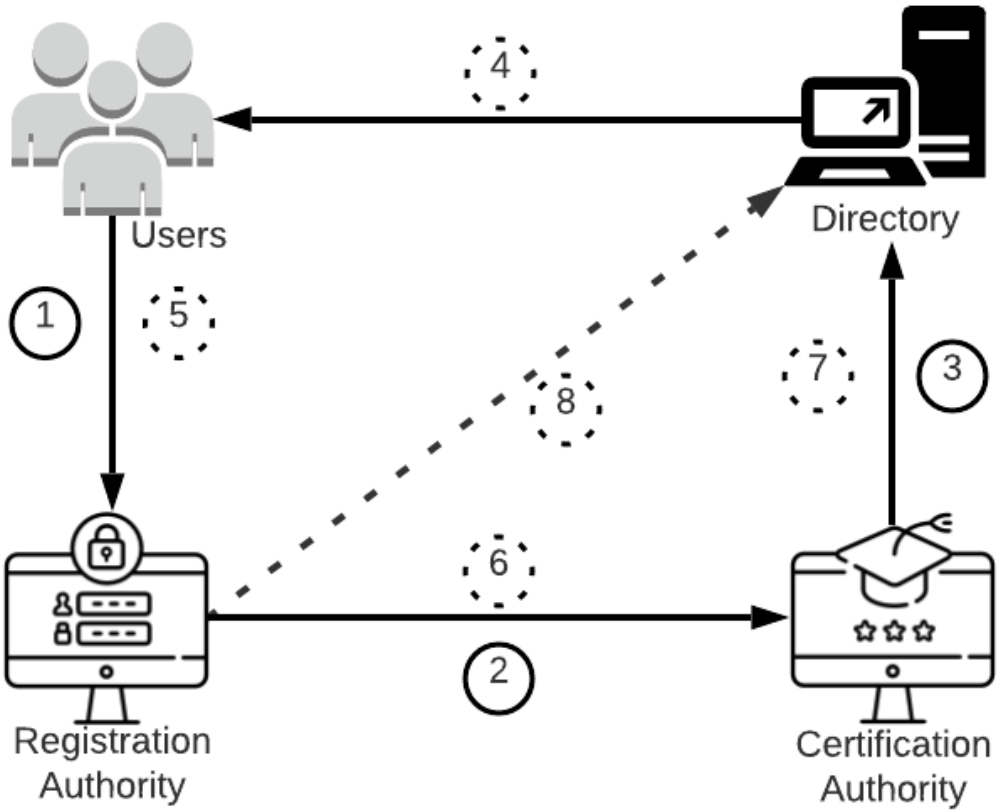
Certificati

- Esiste uno standard per rappresentare l'associazione chiave-titolare attestata da una terza parte: il *certificato di chiave pubblica* (standard ITU-T X.509v3)



Di Andrei Jizdan - opera mia, https://it.wikipedia.org/w/index.php?curid=7795838

Public Key Infrastructure



Initialization

1. Certificate Signing Request (CSR)
2. Approved CSR
3. Certificate Publication

Operation

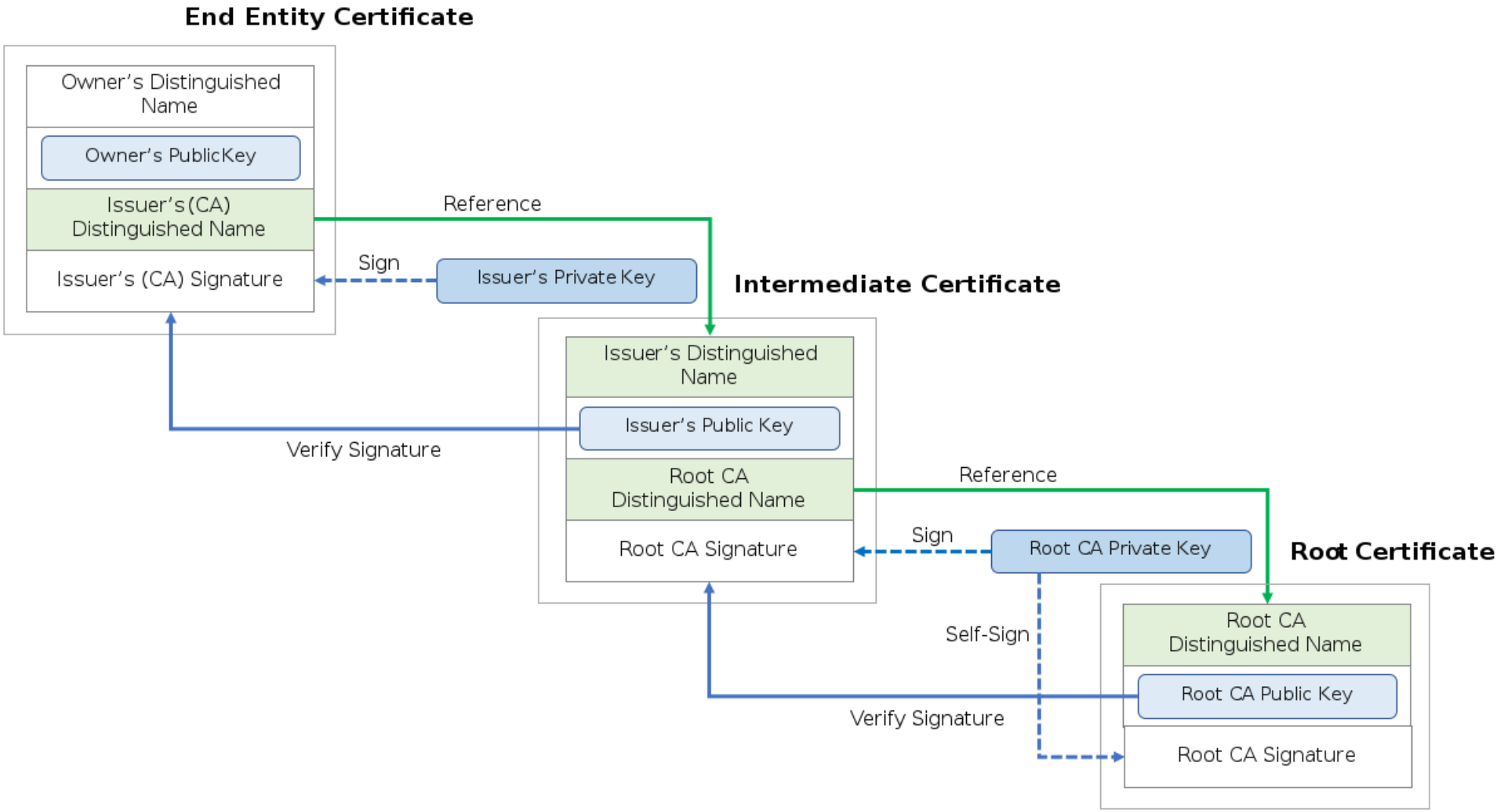
4. Certificate and status Information distribution
5. Revocation/Removal Request (RR)
6. Approved RR
7. Off-line status information publication
8. On-line status information publication

Verifica dei certificati

- Ricevo un messaggio firmato
  - Mi procuro il certificato del mittente
  - con la chiave verifico la firma →
    se ok, messaggio integro e autentico
- E il certificato è integro e autentico?
  - è firmato dalla CA
  - mi procuro il certificato della CA
  - con la chiave verifico la firma →
    se ok, certificato integro e autentico

quando finisce questo iter?

Certificate chain e root CA

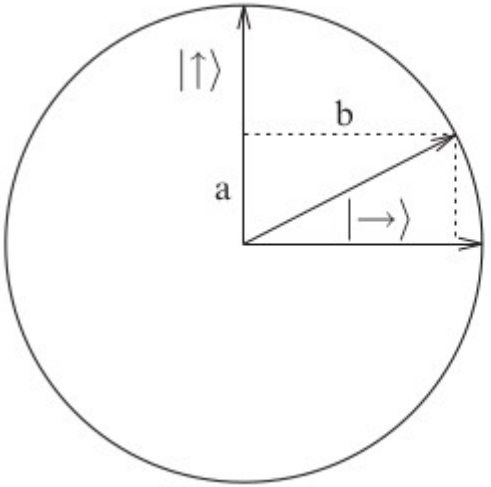


Uno sguardo al futuro

- Quantum key distribution
- In English, sorry!

Qubits

- Let's take a property at the quantum level, e.g. photon polarization
- It can assume any direction in a plane
- The direction can always be expressed as the linear combination of two base states



$$|v\rangle = a|\uparrow\rangle + b|\rightarrow\rangle$$

- **When $a \neq 0$ and $b \neq 0$**, v is a **superposition** of $|\uparrow\rangle$ and $|\rightarrow\rangle$
- there are also n-dimensional (n>2) quantum systems, but they can be modeled using multiple qubits, so let's stick with n=2

Qubits

- For information processing purposes, any physical two-state quantum system is equivalent
- The two elements forming the **basis** are labeled $|0\rangle$ and $|1\rangle$; they encode the bit values 0 and 1
- A qubit can assume any value $|v\rangle = a|0\rangle + b|1\rangle$ with a and b complex numbers such that $|a|^2 + |b|^2 = 1$

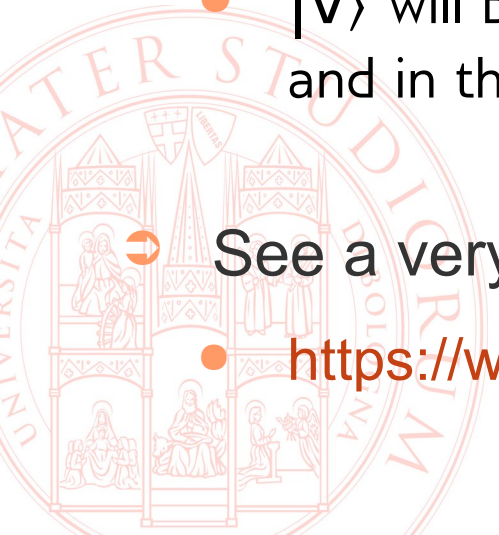
The axiom of measurement

- Any measurement device will have its own basis;
  let's call it $\{|u\rangle , |u^{\perp}\rangle\}$

- "Measuring a qubit" makes sense only with respect to the basis of the device, i.e. we must think of it as $|v\rangle = a|u\rangle + b|u^{\perp}\rangle$

- **The measurement act will change the qubit state**
  - $|v\rangle$ will be measured as $|u\rangle$ with probability $|a|^2$,
    and in that case it will become $|u\rangle$
  - $|v\rangle$ will be measured as $|u^{\perp}\rangle$ with probability $|b|^2$,
    and in that case it will become $|u^{\perp}\rangle$

- See a very good explanation of the "polarizer paradox"
  - https://www.youtube.com/watch?v=zcqZHYo7ONs

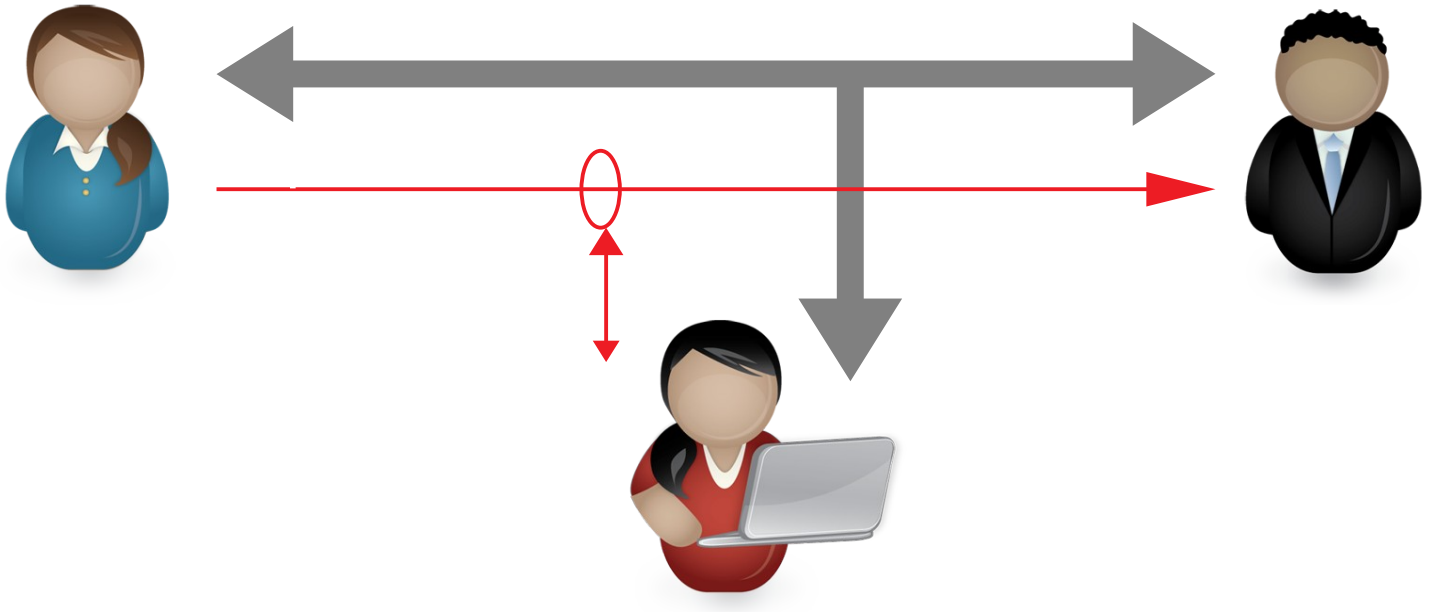


Qubit interpretation

- A qubit can apparently store an unlimited amount of information *but it cannot provide it*
  - measuring it will yield a binary result (one of the basis' components), equivalent to a bit
  - **qubits cannot be cloned**, thus it is impossible to make many measurements on as many copies
- The measurement of a superposition yields probabilistic results
  - tempting interpretation: a qubit is a probabilistic mixture of $|0\rangle$ and $|1\rangle$
  - may be a helpful analogy, but not a strictly correct one
  - in particular it is false that the state is really $|0\rangle$ or $|1\rangle$ and we don't know which
- A qubit always has a definite state, but measuring it in different bases can give deterministic or probabilistic (up to totally random) results, e.g.
  - a photon $|\nearrow\rangle$ can be expressed in the standard basis as $1/\sqrt{2}(|\uparrow\rangle+|\rightarrow\rangle)$
  - measured in the same standard basis, it will turn out as either $|\uparrow\rangle$ or $|\rightarrow\rangle$ **with the same 50% probability**
  - measured in the (Hadamard) basis $\{|\nearrow\rangle,|\nwarrow\rangle\}$, it can be expressed as $1|\nearrow\rangle+0|\nwarrow\rangle$ **so it is not a superposition**, and will deterministically turn out as $|\nearrow\rangle$

Key distribution

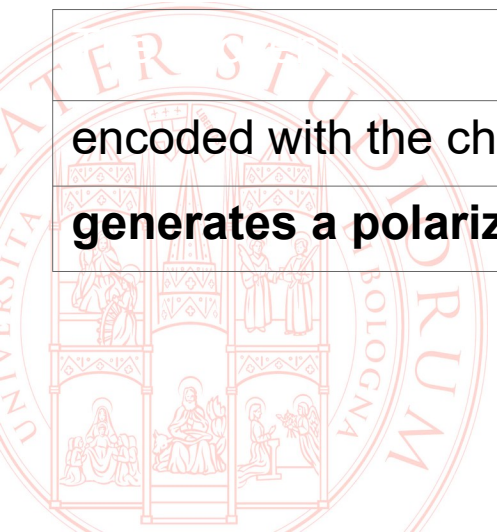
- BB84 (Bennet & Brassard, 1984)
- Establish a secret key between Alice and Bob
  - random sequence of 0s and 1s
  - if it succeeds → high probability that the key is secret
  - no guarantee of success!
- Scenario:



BB84 – setup

- Alice randomly chooses a string of bits and a sequence of bases, one for each bit
  - either the standard basis **S={|↑⟩,|→⟩}**
  - or the Hadamard basis **H={|↗⟩,|↖⟩}**
- Alice encodes the bits with the corresponding bases and sends them to Bob

	0	1	0	0	1	0	1	1	0	1
encoded with the chosen basis:	H	S	H	H	H	S	S	H	S	S
generates a polarized photon:	|↗⟩	|→⟩	|↗⟩	|↗⟩	|↖⟩	|↑⟩	|→⟩	|↖⟩	|↑⟩	|→⟩



BB84 – Alice to Bob

- For each photon, Bob
  - confirms reception to Alice on the classical channel
  - **picks a basis at random to measure it**

0	1	0	1	1	0	1	1	0	1
H	S	H	H	H	S	S	H	S	S
$\|\nearrow\rangle$	$\|\rightarrow\rangle$	$\|\nearrow\rangle$	$\|\nwarrow\rangle$	$\|\nwarrow\rangle$	$\|\uparrow\rangle$	$\|\rightarrow\rangle$	$\|\nwarrow\rangle$	$\|\uparrow\rangle$	$\|\rightarrow\rangle$
H	H	H	H	S	S	H	S	S	H
0	50-50	**0**	**1**	50-50	**0**	50-50	50-50	**0**	50-50

  - **50% chance of choosing the same basis Alice used**
    → getting the same key bit she sent
  - **50% chance of choosing the other one**
    → getting the right or the wrong bit with equal probability

- At the end, Alice and Bob disclose their choice of bases to each other
  - they keep only the bits corresponding to matching bases
  - they compare and discard some of the key bits

BB84 – Eve's chances

- For each photon, **Eve can guess the basis with 50% of success**
  - half of Eve's photons measured with correct basis → bits will be right
  - half of Eve's photons measured with wrong basis → 50% of yielding the right bit
    → 25% of Eve's bits will be wrong
- if she hits the wrong basis, **it will change the photon polarization**

0	1	0	1	1	0	1	1	0	1
H	S	H	H	H	S	S	H	S	S
|↗⟩	|→⟩	|↗⟩	|↖⟩	|↖⟩	|↑⟩	|→⟩	|↖⟩	|↑⟩	|→⟩
H	H	H	S	H	H	S	S	S	H
|↗⟩	|↗⟩ or |↖⟩ 0/1 50-50	|↗⟩	|↑⟩ or |→⟩ 0/1 50-50	|↖⟩	|↗⟩ or |↖⟩ 0/1 50-50	|→⟩	|↑⟩ or |→⟩ 0/1 50-50	|↑⟩	|↗⟩ or |↖⟩ 0/1 50-50
H	H	H	H	S	S	H	S	S	H
0	50-50	**0**	**50-50**	50-50	**50-50**	50-50	50-50	**0**	50-50

- **When Bob measures its photons after Eve's tampering**
  - half of Bob's bits will be kept after comparing bases with Alice
  - half of those will have their polarization changed → 50% of measuring the wrong bit
    → 25% overall probability of Bob measuring a different bit than the one Alice sent
- **Comparing *n* bits, Alice and Bob get a $1/2^{2n}$ probability of Eve's tampering passing undetected**